

Iterative Smoothing Proximal Gradient for Regression with Structured Sparsity

Fouad Hadj-Selem^{*,1}, Tommy Löfstedt^{*,1}, Elvis Dohmatob^{1,2},
Vincent Frouin¹, Mathieu Dubois¹, Vincent Guillemot¹, Edouard
Duchesnay^{*,1} and for the Alzheimer's Disease Neuroimaging
Initiative[†]

¹*NeuroSpin - CEA, Paris-Saclay - France*

²*PARIETAL - Inria Saclay - France*

Abstract: Learning predictive models from high-dimensional brain imaging opens up for possibilities to decode cognitive states or diagnosis/prognosis of a clinical condition/evolution. Structured sparsity is a promising approach which alleviates the risk of overfitting while forcing the solution to adhere to some domain-specific constraints and thus offering new means to identify biomarkers. We consider the problem of optimizing the sum of a smooth convex loss and a non-smooth convex penalty, whose proximal operator is known, and a wide range of complex, non-smooth convex structured penalties such as *e.g.* total variation, or overlapping group lasso. Although many solvers have already been proposed, their practical use in the context of high-dimensional imaging data ($\geq 10^5$ features) remains an open issue. We propose to smooth the structured penalty, since this approach allows a generic framework in which a large range of non-smooth convex structured penalties can be minimized without computing their proximal operators. This is beneficial since the proximal operators are either not known or expensive to compute. The problem can be minimized with an accelerated proximal gradient method, taking advantage of the potential (non-smoothed) sparsity-inducing penalties. As a first contribution we propose an expression of the duality gap to control the convergence of the global non-smooth problem. This expression is applicable to a large range of structured penalties. However, smoothing methods have many limitations that the proposed solver aims to overcome. Therefore, as a second contribution, we propose a continuation algorithm, called *CON-ESTA*, that dynamically generates a decreasing sequence of smoothing parameters in order to maintain the optimal convergence speed towards any globally desired precision. At each continuation step, the aforementioned duality gap provides the current error and thus the next smaller prescribed precision. Given this precision, we propose an expression to calculate the

*Contributed equally and corresponding authors. e-mail: hadjselemfouad@gmail.com; lofstedt.tommy@gmail.com; edouard.duchesnay@cea.fr

[†]Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (<http://www.loni.ucla.edu/ADNI>). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. ADNI investigators include (complete listing available at http://www.loni.ucla.edu/ADNI/Collaboration/ADNI_Authorship_list.pdf)

[‡]Supported by grants from the French National Research Agency: ANR GENIM (ANR-10-BLAN-0128), ANR IA BRAINOMICS (ANR-10-BINF-04), ANR Karametria (ANR-09-BLAN-0332-01) and a European Commission grant: MESCOG (FP6 ERA-NET NEURON 01 EW1207).

optimal smoothing parameter, that minimizes the number of iterations to reach such precision. We demonstrate that CONESTA achieves an improved convergence rate compared to classical (without continuation) proximal gradient smoothing. Moreover, experiments conducted on both simulated and high-dimensional neuroimaging (MRI) data, exhibit that CONESTA significantly outperforms the excessive gap method, ADMM, classical proximal gradient smoothing and inexact FISTA in terms of convergence speed and/or precision of the solution.

1. Introduction

An increasing number of high-dimensional data sets are produced in situations where it is known that the underlying generating model is truly sparse and that there are structured correlations among the many sampled variables. In particular, many data sets of this type are present in the life sciences. Among others, they are encountered in multivariate neuroimaging studies [49], multivariate genome-wide brain-wide association studies [47], and gene expression studies [30].

These kinds of high-dimensional problems are usually addressed with linear multivariate regression or classification methods in conjunction with penalties or constraints based *e.g.* on the ℓ_1 - or ℓ_2 -norms. The properties of these methods are well known and include their ability to deal with large numbers of variables. The elastic net penalty [51] is very popular and is implemented in many machine learning software libraries. However, experimenters often also require that the model solutions adhere to some domain-specific constraints and multivariate methods that can accommodate structured penalties are less widely available [33].

Ideally, the experimenter would like to combine a sparsity-inducing penalty with one or several structured penalties. The hypothesis is that this would improve the interpretability and potentially the accuracy of the model. For example, a group $\ell_{1,2}$ penalty with overlap was used by Silver and Montana [44] in the framework of reduced-rank regression to combine genetic data and imaging traits in order to select single nucleotide polymorphisms (SNPs) that belong to sets of genes involved in the same biological process. The penalty describes groups of SNPs that are linked to specific genes and groups of genes that contribute to specific biological pathways. In a second example, Michel et al. [34] presented a method for predicting behavior from fMRI images using total variation (TV) penalty. Total Variation (TV) penalty forces sparsity on the spatial derivative of the weight map (using a ℓ_{12} penalty), providing the segmentation of weight map into spatially-contiguous parcels with constant values. TV can be combined with sparsity-inducing penalty (ℓ_1), to obtain segmenting properties that extracts predictive regions from a noisy background with zeros [24, 19]. Combining TV with ℓ_1 (features selection) produces the desired foreground-background segmentation of interpretable regions. Finally, [21] combining ℓ_1 +TV with a ℓ_2 penalty prevents inflation of coefficients within selected regions.

In this paper we consider an objective function that consists of the sum of: (i) a smooth convex loss; (ii) a convex non-smooth penalty whose proximal

operator (definition given latter) is known; and (iii) a complex, non-smooth convex structured penalty with unknown proximal operator. Various methods have been proposed to minimize such problems; however, each one requires several auxiliary calculations that limit its application to some particular cases. For instance, proximal algorithms such as the iterative soft-thresholding algorithm (ISTA) [18, 16, 38], the fast, or accelerated ISTA (FISTA) [4, 38] can be used to account for only one of these non-smooth terms alone. Combining two or more non-smooth penalties is expensive, since the proximal operator of the sum of functions is not known in the general case. Besides proximal methods, many other optimization strategies allow minimization of this type of problems when combining several complex or sparsity-inducing penalties. For instance, we cite the algorithms presented in [28, 12, 9, 37, 36], see [19] for a comparison of their performances. Nevertheless, as is explained in Section 2.2, some of them require access to unknown proximal operators of the non-smooth or even the smooth part of the objective function; others require the explicit solutions of some minimization problems that are unsolvable for many useful machine learning models. For example, the generalized forward-backward algorithms [40] deal with the sum of non-smooth functions but require that each one is simple in the sense that its proximity operator is easy to compute.

This paper focuses on proximal methods that may be classified based on two general strategies. The first one involves approximating the proximal operator associated with the convex non-smooth penalties. The second one relies on smoothing all or part of the non-smooth penalties such that the proximal operator becomes as simple as (for example) a soft-thresholding function.

Progress has been made with respect to the first approach, *i.e.* the approximation of proximal operators using inexact proximal operators in FISTA [43, 34]. Michel et al. [34] approximated the proximal operator of the TV constraint. This procedure can also be applied to more complex composite non-smooth constraints, such as TV combined with the ℓ_1 penalty. Unfortunately, as we will see below, approximating the corresponding proximal operator causes the computational time to increase dramatically.

We therefore choose to investigate the second approach, *i.e.* smoothing the non-smooth penalty with unknown proximal operator, while keeping the exact ℓ_1 constraint and using a proximal gradient method to minimize the whole function. This approach, *smoothing proximal gradient method (SPG)*, has been proposed by Chen et al. [15]. The authors combined an ℓ_1 penalty with a non-smooth group-wise $\ell_{1,2}$ penalty. They replaced the group $\ell_{1,2}$ penalty with a smooth function using Nesterov's smoothing technique [36]. The resulting objective function was then minimized using FISTA. It is worth noting that Nesterov's smoothing procedure involves a smoothing parameter μ and that the choice of μ reflects a trade-off. The convergence rate of FISTA depends on μ : the smaller μ is, the slower FISTA converges. On the other hand, a small μ guarantees a good approximation of the smoothed penalty.

Savchynskyy et al. [42] addressed this issue and proposed an adaptive diminishing smoothing scheme for μ . The authors proposed a continuation method that updates the regularization parameter at the beginning of each continuation

iteration to increase the precision of the solution. However, they imposed a full regularization of the non-smooth part that prevents true sparsity. Moreover, their method is not straightforwardly applicable to structured linear regression. In fact, they considered a well-studied class in linear programming and involve many prior results that are specific to their problem and the imposed full smoothing. In a benchmarking of solvers [19], authors found that, for a logistic loss, continuation combined with smoothing had the best convergence speed (in time) followed by ADMM.

Another theoretically sound continuation method is Nesterov's excessive gap method [36]. Nesterov showed that it achieves the best possible convergence rate towards the solution of the original optimization problem. Nevertheless, it requires to have access to the explicit solution of some auxiliary minimization problems that are generally not available. Thus, the excessive gap method suffers from some major limitations: restriction to the least-squares loss and full regularization of the non-smooth part of the objective function, thereby loosing any sparsity-inducing properties.

This brief review outlined the potential shortcomings of state-of-the-art methods. However, it suggested that Nesterov's smoothing combined with a continuation approach is a promising track to follow.

This paper aims to overcome those limitations, and we propose a new continuation algorithm for the minimization of an objective function containing non-smooth convex terms whose proximal operator is unknown. We call this algorithm CONESTA, which stands for **C**Ontinuation with **N**esterov smoothing in a **S**hrinkage-**T**hresholding **A**lgorithm. The proposed algorithm addresses the limitations of the state-of-the-art methods through three main contributions:

- (i) The continuation scheme. It dynamically generates a sequence of decreasing precisions ε^i , is calculated from an estimation of the current error (see duality gap in point iii). Such a continuation scheme ensures the convergence towards a globally desired final precision, ε , which is the only parameter that the user needs to provide.
- (ii) Given a precision level, ε^i , we propose an expression for the value of the smoothing parameter, μ^i , that minimizes the number of iterations required for convergence to the specified precision, ε^i .
- (iii) All error estimates are based on the duality gap: First, the duality gap is used as a stopping criterion for the outer (global) continuation. Second, the duality gap is also used to stop the inner application of FISTA, as soon as the prescribed precision ε^i has been reached. This prevents unnecessary convergence toward the approximated (*i.e.*, smoothed) objective function. Finally, the duality gap is used to dynamically determine the prescribed precision, ε^i , with respect to the current error estimation of the true objective function. Those two points are crucial when considered together with item (ii): if the duality gap indicates that we are close to the solution, the precision, ε^i , will be small, and thus, thanks to item (ii), the smoothing will be optimally adapted to the current situation.

The paper is structured as follows: After a detailed review of the state-of-the-

art methods in Section 2, we present CONESTA in Section 3 and the three main contributions that constitute it. In Section 4, we present its application to linear regression with both elastic net and TV penalties. We conducted validation studies on both a simulated data set, Section 5, and on a high-dimensional experimental 3D neuroimaging data set, Section 6.

The convergence speed of CONESTA was compared to that of several state-of-the-art minimization algorithms: the excessive gap method [36], the alternating direction method of multipliers (ADMM) [9], the fast iterative soft-thresholding algorithm (FISTA) [4], the *inexact* fast proximal gradient algorithm [43] and the smoothing proximal gradient method, denoted “FISTA with Chen’s μ ” below [15].

We limited ourselves to these approaches in order to maintain readability of the paper and because these methods achieve optimal convergence speed. The comparisons clearly demonstrate that CONESTA outperformed the other state-of-the-art minimization algorithms in this study, when solving a linear regression problem with elastic net and TV penalties on both experimental and simulated data sets. We also demonstrate the relevance of TV for recovering stable and meaningful predictive support on neuroimaging data.

2. Background

Section 2.1 introduces to the general minimization problem and its application to penalized linear regression models. Section 2.2 presents the current state-of-the-art optimization algorithms and their limitations that justify the relevance of the proposed contribution. Finally, Section 2.3 describes Nesterov’s smoothing technique and Section 2.4 shows its use together with FISTA, which is the base on which we built the CONESTA algorithm.

2.1. Problem statement

The goal is to minimize the general optimization problem:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) = \min_{\beta \in \mathbb{R}^p} \{g(\beta) + \gamma s(\beta) + \kappa h(\beta)\}, \quad (2.1)$$

where β is the vector of parameters to be estimated, g is a differentiable convex function with a Lipschitz continuous gradient, h is a convex non-differentiable function with a known proximal operator and s is an additional convex non-differentiable structured penalty whose proximal operator is not necessarily known or is difficult to express or compute. The real scalars $\kappa \geq 0$ and $\gamma \geq 0$ are regularization parameters. We further denote¹

$$f^* = \min_{\beta} f(\beta) \quad \text{and} \quad \beta^* = \arg \min_{\beta} f(\beta).$$

¹The last equality should be understood in the sense $\beta^* \in \arg \min_{\beta} f(\beta)$, in case there are more than one minimizers (this may happen in the absence of strong convexity).

This class of optimization problems is very general and includes classical models such as the lasso, ridge regression and elastic net, the perhaps less common TV [10] and group lasso ($\ell_{1,2}$) [50, 39] models, and combinations of these.

Consider a data set (\mathbf{X}, \mathbf{y}) , where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is a matrix of n input vectors of dimension p and $\mathbf{y} \in \mathbb{R}^n$ denotes the corresponding n -dimensional response vector. In this paper we utilize penalized linear regression functions, *i.e.* where the loss function is $\frac{1}{2} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$, to illustrate the application of such optimization problems. We will additionally also focus on both an elastic net and a “group” penalty. The function we wish to minimize is therefore

$$f(\boldsymbol{\beta}) = \underbrace{\frac{1}{2} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2}_{g(\boldsymbol{\beta})} + \underbrace{\gamma \sum_{G \in \mathcal{G}} \|\mathbf{A}_G \boldsymbol{\beta}_G\|_2}_{s(\boldsymbol{\beta})} + \underbrace{\kappa \|\boldsymbol{\beta}\|_1}_{h(\boldsymbol{\beta})}, \quad (2.2)$$

where $\lambda \geq 0$, the function $h(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$ is a sparsity-inducing penalty whose proximal operator is known, $s(\boldsymbol{\beta}) = \sum_{G \in \mathcal{G}} \|\mathbf{A}_G \boldsymbol{\beta}_G\|_2$ is a penalty on the structure of the variables, with \mathcal{G} being a set of potentially overlapping groups (a group G has size $|G|$) of variables and \mathbf{A}_G a linear operator on $\mathbb{R}^{|G|}$ (an example of such an operator is detailed in Section 4). TV and group $\ell_{1,2}$ are two examples of such “group” penalties.

2.2. Relation to prior works

Inexact proximal-gradient Schmidt, Le Roux and Bach [43] gave a general sufficient condition for applying *inexact* proximal gradient algorithms where the proximal operator (or the gradient) is approximated. They established a sufficient condition for the approximation of the unknown proximal operator to be applied at each step of the proximal gradient algorithm such that the optimal convergence speed is maintained. This method could be used in the application of complex penalties such as total variation or group lasso in linear regression. The approximation is performed by solving a subproblem to find the proximal operator. Details are provided in Section SM 1.2 of the supplemental article [26]. [46, 19] implemented this method in the case of TV-regularized problems (like the ones considered in the current paper), and showed that it outperforms splitting and primal-dual methods. However, at least in the case of for high-dimensional problems (like those we are concerned with here), solving a subproblem like the one solved in approximating the prox of g is possibly a time-consuming process. Indeed, to ensure convergence, the prescribed precision of the inner loop that solves the subproblem must decrease faster than $\mathcal{O}(1/k^{2+\delta})$ with $\delta > 0$ (where k is the number of FISTA iterations) [43, Proposition 1]. Thus, a precision $\varepsilon_{\text{prox}} \approx 10^{-4}$ is required already in the computation of the prox of g after a hundred iterations of the outer FISTA loop. Such precision requires hundred iterations of the inner FISTA loop, which operates only on $\boldsymbol{\beta}$. Therefore, inexact proximal gradient algorithm is adapted for problems

with moderate dimensionality where the required precision will not involve a large number of iterations of outer FISTA loop [20]. However, this is a crucial issue with very high dimensional problems such as neuroimaging, or genetic data analysis. With a loss of generality, this problem can be alleviated by developing a specific optimized code to speed-up the computation of the proximal operator of given structured penalty. In contrast, the method that we propose here, CONESTA, does not suffer from this limitation since the approximation is not found numerically, and so no inner loop is needed. Moreover, the smoothing approach offers a generic framework in which a large range of non-smooth convex structured penalties can be minimized without computing their proximal operators. Moreover, the conducted comparison study demonstrated that CONESTA is much faster than the *inexact* proximal gradient method in terms of the convergence time.

Primal-dual Chambolle and Pock [13] proposed a general primal-dual method that minimizes a loss function with two different penalties, such as *e.g.* TV and ℓ_1 [25]. They consider a function in the form $g(Kx) + h(x)$ where K is a linear operator and proved an optimal convergence rate for such non-smooth objective functions. However they assume to have access to the proximal operators of both the smooth function, g , and the non-smooth one, h . This is the main shortcoming of this method, as it would require the approximation of said proximal operators when they are not available, like in the case with logistic regression, for instance. It would then require an inexact approach like the one for proximal gradient methods developed by Schmidt, Le Roux and Bach [43] and discussed above. To the best of our knowledge, the inexact issue for this method is an open problem.

Excessive gap Nesterov [36] presented an optimal primal-dual method called the excessive gap method, or the excessive gap technique. As we will show later, this method can be used to minimize Equation 2.2 with optimal convergence rate. Nevertheless it shares the same shortcomings as the method by Chambolle and Pock [13]. In fact, a necessary step in the excessive gap algorithm is the computation of $\hat{\beta}$, the primal variable that corresponds to a particular dual variable, \mathbf{u} (see Section SM 3.1 in the supplemental article [26]). As far as the authors know, there is no explicit expression for this function when dealing with the general case of any convex loss function. An explicit application of the excessive gap method can be found in [14] where the authors applied it to canonical correlation analysis with group and fused lasso penalties. A problem with the excessive gap method (illustrated in [14] and reiterated below) is that it imposes a smoothing of all non-smooth parts, and in particular of the ℓ_1 penalty. This implies that the found solution will not be strictly sparse.

Smoothing proximal gradient Nesterov's smoothing can also be used in conjunction proximal gradient algorithm [15]. The main issue of this approach is that accurate solution, with small smoothing, involve a slow convergence. In

Section 2.3 we provide more details on this approach since it is precisely the contribution of this paper.

ADMM The alternating direction method of multipliers [9] is commonly used to minimize the sum of two convex functions. It can be adapted to the loss function of interest in this paper, Equation 2.2, but has some drawbacks in this context. In particular, it suffers from the same shortcoming as the method discussed above by Chambolle and Pock [13]. In fact, when dealing with a smooth loss function that has an unknown proximal operator, the proximal operator would have to be approximated numerically. Another computational limitation of ADMM when working with the ordinary least squares (OLS) loss and the ℓ_1 -norm and TV penalties, is that each update involves solving two linear systems, or, equivalently, computing the inversion of two large $p \times p$ linear operators, possibly ill-conditioned. In the case of total variation and ℓ_1 penalties, this is not a problem, because of the particular form of the resulting linear operator. Yet for more general linear operators, solving the associated linear systems would quickly become intractable, in particular for large p . Moreover the regularization parameter, ρ , in the associated augmented Lagrangian function is difficult to set (this is still an open problem), and even though ADMM converges for any value of this parameter, under mild conditions, the convergence rate depends heavily on it. We have employed some heuristics for selecting this parameter. This issue, mentioned in [19, 43], is discussed Section SM 3.2 in the supplemental article [26] along with the details of our implementation of ADMM as described in [48]. For the sake of method comparison, we had to generalize it slightly and have added an ℓ_1 penalty. We note that this kind of modification would not be as trivial with general complex penalties.

PRISMA PRISMA [37] is a continuation algorithm for minimizing a convex objective function that decomposes into three parts: a smooth part, a simple non-smooth Lipschitz continuous part, and a simple non-smooth non-Lipschitz continuous part. They use a smoothing strategy similar to that used in this paper. The main limitation is that the two different penalties have to be simple such that their proximal operators are explicit (see Algorithm 1 in [37]). Thus, as there is no inexact approach that allows to approximate any unknown proximal operator, while preserving the convergence, we can not apply PRISMA in a rigorous way when dealing with group lasso or total variation.

Our proposed continuation algorithm addresses the two aforementioned deficiencies. Indeed, CONESTA (i) is relevant in the context of any smooth convex loss function because it only requires the computation of the gradient and (ii) estimates weights that are strictly sparse because it does not require to smooth the sparsity-inducing penalties. Additionally, CONESTA does not require solving any linear systems, or inverting any matrices, and can easily be applied with a variety of convex smooth loss functions and many different complex convex penalties. Similar ideas are also explored in a preliminary work [45].

2.3. Nesterov's smoothing technique

Our proposition is based on the assumption that many structured sparsity-inducing penalties can be reformulated into a simple linear transformation of β via the dual norm. It turns out that this assumption is often true, which can be proved by using results from [7] and is exemplified in [35, Section 4] and [15, Section 3]. Based on this assumption Nesterov's smoothing technique can be applied on all of the non-smooth convex functions we are dealing with, if they respect the condition presented in Equation 2.4. Using the duality concept between the ℓ_q and $\ell_{q/q-1}$ norms (see [7] for instance), we can establish the dual form of $s(\beta)$:

$$s(\beta) = \sum_{G \in \mathcal{G}} \|A_G \beta\|_2 = \sum_{G \in \mathcal{G}} \max_{\|\alpha_G\|_2 \leq 1} \langle \alpha_G | A_G \beta \rangle, \quad (2.3)$$

where $\alpha_G \in \mathbb{R}^{|G|}$ is a vector of auxiliary variables associated with A_G .

Let $\alpha = [\alpha_{G_1}^\top, \dots, \alpha_{G_{|\mathcal{G}|}}^\top]^\top$ be a vector of length $\sum_{G \in \mathcal{G}} |G|$ that is the concatenation of the α_G . Similarly, we combine the linear operators A_G into a concatenated one, denoted $A \in \mathbb{R}^{\sum_{G \in \mathcal{G}} |G| \times p}$. We thus obtain a more compact representation of $s(\beta)$ by

$$s(\beta) = \max_{\alpha \in \mathcal{K}} \langle \alpha | A \beta \rangle, \quad (2.4)$$

where $\alpha \in \mathcal{K} = \{\alpha = [\alpha_{G_1}^\top, \dots, \alpha_{G_{|\mathcal{G}|}}^\top]^\top : \|\alpha_G\|_2 \leq 1, \forall G \in \mathcal{G}\}$ and the set \mathcal{K} is the Cartesian product of unit balls in Euclidean space and, therefore, a compact convex set.

This expression of $s(\beta)$ is a sufficient condition to apply Nesterov's smoothing technique. Nesterov's smoothing technique is then applied as

$$\begin{aligned} s_\mu(\beta) &= \max_{\alpha \in \mathcal{K}} \left\{ \langle \alpha | A \beta \rangle - \frac{\mu}{2} \|\alpha\|_2^2 \right\} \\ &= \langle \alpha_\mu^*(\beta) | A \beta \rangle - \frac{\mu}{2} \|\alpha_\mu^*(\beta)\|_2^2, \end{aligned} \quad (2.5)$$

for all $\beta \in \mathbb{R}^p$, where μ is a positive real smoothing parameter and

$$\alpha_\mu^*(\beta) = \arg \max_{\alpha \in \mathcal{K}} \left\{ \langle \alpha | A \beta \rangle - \frac{\mu}{2} \|\alpha\|_2^2 \right\}. \quad (2.6)$$

The variable $\alpha_\mu^*(\beta)$ can be computed by projecting it onto the compact \mathcal{K} , as stated in Equation 2.11.

The following theorem [35, Theorem 1] presents Nesterov's smoothing when applied to a class of complex penalties.

Theorem 1 (from Nesterov's Theorem [35]). *Let the function s_μ be Nesterov's smooth transform of a convex function s . We state three results:*

(i) *First, s_μ is convex and differentiable with gradient*

$$\nabla s_\mu(\beta) = A^\top \alpha_\mu^*(\beta), \quad (2.7)$$

(ii) Second, the gradient is Lipschitz continuous with constant

$$L(\nabla s_\mu) = \frac{\|\mathbf{A}\|_2^2}{\mu}, \text{ where } \|\mathbf{A}\|_2 = \max_{\mathbf{v} \neq 0} \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} \quad (2.8)$$

is the matrix spectral norm of \mathbf{A} .

(iii) Finally, we have

$$\lim_{\mu \rightarrow 0} s_\mu(\boldsymbol{\beta}) = s(\boldsymbol{\beta}), \quad (2.9)$$

for all $\boldsymbol{\beta} \in \mathbb{R}^p$.

The proof of Equation 2.9 is based on the following Lemma.

Lemma 1. Let s_μ be Nesterov's smooth transform of s . If $M = \max_{\boldsymbol{\alpha} \in \mathcal{K}} \|\boldsymbol{\alpha}\|_2^2/2$, then

$$s_\mu(\boldsymbol{\beta}) \leq s(\boldsymbol{\beta}) \leq s_\mu(\boldsymbol{\beta}) + \mu M, \quad (2.10)$$

for all $\boldsymbol{\beta} \in \mathbb{R}^p$.

In addition, we have the following result that allows us to practically compute the dual variable.

Lemma 2 (Projection formulation of $\boldsymbol{\alpha}_\mu^*$). Using the same notation as above, we can express $\boldsymbol{\alpha}_\mu^*$ as a projection of $\frac{1}{\mu}\mathbf{A}\boldsymbol{\beta}$ onto the compact space \mathcal{K} , i.e.

$$\boldsymbol{\alpha}_\mu^*(\boldsymbol{\beta}) = \text{proj}_{\mathcal{K}} \left(\frac{1}{\mu} \mathbf{A}\boldsymbol{\beta} \right). \quad (2.11)$$

2.4. FISTA with Nesterov's smoothing

Consider a function s , whose proximal operator is not known, or too expensive to compute. This function can instead be smoothed using Nesterov's technique. By this smoothing, we obtain a new optimization problem that is closely related to the one given in Equation 2.1, namely

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f_\mu(\boldsymbol{\beta}) \equiv \overbrace{g(\boldsymbol{\beta}) + \gamma s_\mu(\boldsymbol{\beta})}^{\text{smooth}} + \overbrace{\kappa h(\boldsymbol{\beta})}^{\text{non-smooth}}, \quad (2.12)$$

where s_μ is Nesterov's smoothing of the function s . Additionally, we use the notation $\boldsymbol{\beta}_\mu^* = \arg \min_{\boldsymbol{\beta}} f_\mu(\boldsymbol{\beta})$ to denote the regression vector that minimises f_μ . Problems such as those in Equation 2.12 can be efficiently solved by proximal gradient algorithms such as FISTA [4]. Next we provide all the required quantities to run FISTA, provided in Algorithm 1:

- A random starting vector: $\boldsymbol{\beta}^0 \in \mathbb{R}^p$.
- A fixed smoothing parameter μ .
- A prescribed final precision of the smoothed problem: $\varepsilon > 0$.
- A linear operator \mathbf{A} , that encodes the structure of the complex penalty and the corresponding regularization parameter, γ .

- The gradient of the smooth loss plus the gradient of the smoothed complex penalty (Equation 2.7), *i.e.* $\nabla(g + \gamma s_\mu)(\cdot)$. This function implicitly provides the data \mathbf{X}, \mathbf{y} and the ℓ_2 penalty parameter λ .
- The proximal operator of the non-smooth part, as defined Section SM 1.2 in the supplemental article [26], *i.e.* $\text{prox}_{t_\mu \kappa h}(\cdot)$. The proximal operator implicitly provides the parameter κ of the ℓ_1 penalty.
- The Lipschitz constant of the gradient of the loss function, *i.e.* $L(\nabla(g))$. It is used to compute the step size t_μ (Equation 2.14).
- The stopping criterion (defined as STOPPINGCRITERION) will be either

$$\text{STOPPINGCRITERION} = \begin{cases} \frac{1}{t_\mu} \|\beta^k - \beta^{k-1}\|_2, & \text{if default criterion,} \\ \widetilde{\text{GAP}}_\mu(\beta^k), & \text{if duality gap.} \end{cases} \quad (2.13)$$

For the instant, the reader need only be concerned with the default stopping criterion.

Beck and Teboulle [4] stated that t_μ must be smaller than or equal to the reciprocal of the Lipschitz constant of the gradient of the smooth part, namely of $g + \gamma s_\mu$. Using Theorem 1 to obtain the Lipschitz constant, and considering the fact that we want to take the largest possible step, we choose

$$t_\mu = \frac{1}{L(\nabla(g)) + \gamma \frac{\|\mathbf{A}\|_2^2}{\mu}}. \quad (2.14)$$

This value of t_μ is sufficient for the convergence of FISTA [4], provided in Algorithm 1.

Algorithm 1 The fast iterative shrinkage-thresholding algorithm (FISTA) with fixed μ

Input: $\beta^0, \mu, \varepsilon, \mathbf{A}, \gamma, \nabla(g + \gamma s_\mu)(\cdot), \text{prox}_{t_\mu \kappa h}(\cdot), L(\nabla(g)), \text{STOPPINGCRITERION}$

Output: β^k such that $f_\mu(\beta^k) - f_\mu(\beta^*) < \varepsilon$

```

1:  $\beta^1 = \beta^0$ 
2:  $k = 2$ 
3:  $t_\mu = \left( L(\nabla(g)) + \gamma \frac{\|\mathbf{A}\|_2^2}{\mu} \right)^{-1}$ 
4: loop
5:    $\mathbf{z} = \beta^{k-1} + \frac{k-2}{k+1} (\beta^{k-1} - \beta^{k-2})$ 
6:    $\beta^k = \text{prox}_{t_\mu \kappa h}(\mathbf{z} - t_\mu \nabla(g + \gamma s_\mu)(\mathbf{z}))$ 
7:   if STOPPINGCRITERION  $\leq \varepsilon$  then
8:     break
9:   end if
10:   $k = k + 1$ 
11: end loop
```

FISTA is thus applied to a smoothed version of the original function. It will converge towards β_μ^* , the minimum of f_μ . Using Equation 2.10, we can prove that, for a prescribed precision $\varepsilon > 0$, selecting μ smaller than $\varepsilon/\gamma M$ allows us to achieve the precision $f(\beta^k) - f(\beta^*) \leq \varepsilon$ for f . For example, using $\mu = \varepsilon/2\gamma M$ and the precision $\varepsilon/2$ as a stopping criterion in Algorithm 1, as was done by

Chen et al. [15], would enable us to achieve the precision ε for f (see Section 3.2 below).

Since we do not make the assumption that g in Equation 2.1 is strongly convex, the convergence rate of FISTA, when applied to Equation 2.12, is governed by the expression [4]

$$f_\mu(\beta^k) - f_\mu(\beta_\mu^*) \leq \frac{2}{t_\mu(k+1)^2} \|\beta^0 - \beta_\mu^*\|_2^2, \quad (2.15)$$

where k is the iteration counter.

3. The CONESTA algorithm

This section presents the algorithm CONESTA (Continuation with Nesterov's Smoothing in a Shrinkage-Thresholding Algorithm) that smooths a part of the non-smooth function. The maximal convergence speed towards the minimum of the original (non-smoothed) function is obtained by a continuation procedure that systematically decreases the smoothing parameter.

3.1. Continuation

The step size t_μ used in Algorithm 1 depends on the smoothing parameter μ . They are linked to one another and vary in the same way, although they drive different aspects of the algorithm's convergence. Indeed, high precision, with a small μ , will lead to a slow convergence (small t_μ). Conversely, poor precision (large μ) will lead to rapid convergence (large t_μ). There is therefore a trade-off between speed and precision [6, 41, 42].

It seems natural that we should perform successive runs of an iterative optimization algorithm with successively smaller values of μ in every run. Each run benefits from the starting point provided by the previous loop, and from the *large* (compared to subsequent runs) smoothing parameter which generates a larger step size and thus a smaller number of iterations. To achieve this, we would like to infer an optimal reduced μ after each FISTA loop.

We propose a continuation process in which each FISTA loop achieves a precision ε^i that decreases dynamically depending on the current estimation of the error. This sequence converges to the prescribed precision, ε , initially chosen by the user. Once ε^i is fixed, we calculate an appropriate smoothing parameter μ^i to achieve this precision with the least number of iterations. We detail here these main steps of the CONESTA algorithm starting from the current β^i :

1. Estimate the current error: $f(\beta^i) - f(\beta^*)$. This estimation is found using the duality gap (see Section 3.3).
2. Compute the desired accuracy ε^{i+1} for the next FISTA loop in order to achieve a smaller error, $f(\beta^{i+1}) - f(\beta^*)$. To obtain ε^{i+1} , the current accuracy at the i th iteration, ε^i , is decreased by a factor τ .

3. Compute the smoothing parameter μ^{i+1} that gives the smallest upper bound on the number of iterations required to achieve the precision ε^{i+1} , i.e. such that $f(\beta^{i+1}) - f(\beta^*) \leq \varepsilon^{i+1}$, in the least number of iterations possible. See Section 3.2.
4. Run FISTA on $f_{\mu^{i+1}}$ to obtain β^{i+1} .

We then iterate until the error $f(\beta^i) - f(\beta^*) < \varepsilon$, for a prescribed final stopping precision $\varepsilon > 0$.

The remainder of this section is organized as follows: First, we describe how to obtain an optimal value of μ that minimizes the upper bound on the number of iterations required to achieve a prescribed precision ε when minimizing f_μ . Next, we establish an upper bound for the duality gap in Section 3.3. Finally, we present CONESTA.

3.2. Determining the optimal smoothing parameter

The smoothed function, f_μ , as mentioned in Equation 2.10 and Equation 2.12, provides upper and lower bounds on the original function, f , such that

$$f_\mu \leq f \leq f_\mu + \gamma\mu M, \quad (3.1)$$

with $M = \max_{\alpha \in \mathcal{K}} \frac{1}{2} \|\alpha\|_2^2$. The following theorem provides the μ that achieves the prescribed precision ε with the least number of iterations.

Theorem 2 (Optimal smoothing parameter μ). *For any given $\varepsilon > 0$, selecting the smoothing parameter μ as*

$$\mu_{opt}(\varepsilon) = \frac{-\gamma M \|\mathbf{A}\|_2^2 + \sqrt{(\gamma M \|\mathbf{A}\|_2^2)^2 + ML(\nabla(g)) \|\mathbf{A}\|_2^2 \varepsilon}}{ML(\nabla(g))}. \quad (3.2)$$

minimizes the worst case bound on the number of iterations required to achieve the precision

$$f(\beta^k) - f(\beta^*) < \varepsilon. \quad (3.3)$$

The proof is provided in Section SM 2.1 in the supplemental article [26].

3.3. Duality gap

Duality formulations are often used to control the achieved precision when minimizing convex functions. They can be used to provide an estimation of the error $f(\beta) - f(\beta^*)$, for any β , without knowing the minimum $f(\beta^*)$, which we never know in practice. The duality gap is the cornerstone of the CONESTA algorithm. Indeed, it is used three times:

1. In the i th CONESTA iteration, as a way to estimate the current error $f(\beta^i) - f(\beta^*)$. The error will be estimated using the gap of the smoothed problem. This is justified in Section 3.5. This value is then used to deduce all the other parameters for the next application of FISTA. The next desired precision and the smoothing parameter, μ^i , are derived from this value.

2. As the stopping criterion in the inner FISTA loop. The criterion will be such that FISTA will stop as soon as the current precision is achieved using the current smoothing parameter, μ^i . This prevents unnecessary convergence toward the approximated (smoothed) objective function.
3. Finally, as the global stopping criterion within CONESTA. This will guarantee that the obtained approximation of the minimum, β^i , at convergence, satisfies $f(\beta^i) - f(\beta^*) < \varepsilon$.

We first establish an expression of the duality gap for the problem in Equation 2.2. Next, we consider the smoothed version in Equation 2.12.

The Fenchel duality can be used as in [32] to rewrite the objective function in Equation 2.2 as

$$f(\beta) = \underbrace{\frac{1}{2} \|\mathbf{X}\beta - \mathbf{y}\|_2^2}_{l(\mathbf{X}\beta)} + \underbrace{\frac{\lambda}{2} \|\beta\|_2^2 + \kappa \|\beta\|_1 + \gamma \max_{\alpha \in \mathcal{K}} \langle \alpha | A\beta \rangle}_{\psi(\beta)} \equiv l(\mathbf{X}\beta) + \psi(\beta), \quad (3.4)$$

where we used Nesterov's assumption from Equation 2.4 on our penalty. Note that the squared loss is expressed as a function of $\mathbf{X}\beta$ using $l(\mathbf{x}) \equiv \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$. The rules from classical Fenchel duality [8] provide the duality gap for the problem in Equation 3.4 at the current value of β^k

$$\text{GAP}(\beta^k) \equiv f(\beta^k) + l^*(\sigma(\beta^k)) + \psi^*(-\mathbf{X}^\top \sigma(\beta^k)), \quad (3.5)$$

where l^* and ψ^* are the Fenchel conjugates of l and ψ , respectively. It is straightforward to show that l^* can be expressed as $l^*(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|_2^2 + \langle \mathbf{x}, \mathbf{y} \rangle$. Following Mairal [32], the dual variable associated to a given β^k can be computed as $\sigma(\beta^k) \equiv \nabla l(\mathbf{X}\beta^k)$. The duality gap is finite, it vanishes at the minimum and provides an estimate of the difference with the optimal value of the objective function. The duality gap has the following properties:

$$\begin{aligned} \text{GAP}(\beta^k) &\geq f(\beta^k) - f(\beta^*) \geq 0; \\ \text{GAP}(\beta^*) &= 0. \end{aligned} \quad (3.6)$$

This requires the calculation of l^* and ψ^* . However, to the best of our knowledge, there is no explicit expression for ψ^* when using a complex penalty such as total variation or group lasso. Therefore, we use an approximation that maintains the properties stated in Equation 3.6. We use Nesterov's assumption and rewrite $\psi(\beta)$ as

$$\begin{aligned} \psi(\beta) &= \frac{\lambda}{2} \|\beta\|_2^2 + \kappa \|\beta\|_1 + \gamma \max_{\alpha \in \mathcal{K}} \langle \alpha | A\beta \rangle \\ &= \max_{\alpha \in \mathcal{K}} \left\{ \frac{\lambda}{2} \|\beta\|_2^2 + \kappa \|\beta\|_1 + \gamma \langle \alpha | A\beta \rangle \right\} \\ &= \max_{\alpha \in \mathcal{K}} \psi_\alpha(\beta) \end{aligned} \quad (3.7)$$

Then, we approximate the Fenchel conjugate of ψ by that of $\psi_k \equiv \psi_{\alpha(\beta^k)}$, which is the local approximation of ψ at the current β^k . Next, we define an

approximation for the gap by explicitly computing ψ_k^* . When $\lambda = 0$, the function ψ_k^* may be infinite at some $\boldsymbol{\sigma}(\boldsymbol{\beta}^k)$. Thus, we slightly change it in order to obtain feasible values. An explicit expression for the duality gap is presented in the following theorem.

Theorem 3 (Duality gap for the non-smooth problem). *Let λ, κ and γ be non-negative real numbers. The following estimation of the gap satisfies Equation 3.6:*

$$\widetilde{\text{GAP}}(\boldsymbol{\beta}^k) \equiv f(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \psi_k^*(-\mathbf{X}^\top \boldsymbol{\sigma}^k) \quad (3.8)$$

with $\psi_k^*(\mathbf{v})$ being the Fenchel conjugate:

$$\psi_k^*(\mathbf{v}) \equiv \begin{cases} \frac{1}{2\lambda} \sum_{j=1}^p \left([|v_j - s_j^k| - \kappa]_+ \right)^2 & \text{if } \lambda > 0, \\ 0 & \text{if } \lambda = 0, \end{cases}$$

and

$$\boldsymbol{\sigma}^k \equiv \begin{cases} \nabla l(\mathbf{X}\boldsymbol{\beta}^k) & \text{if } \lambda > 0 \\ (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}(\mathbf{k}^k - \mathbf{s}^k) & \text{if } \lambda = 0. \end{cases}$$

The vectors $\mathbf{s}^k, \mathbf{k}^k \in \mathbb{R}^p$ are given by

$$\mathbf{s}^k = \gamma(\mathbf{A}^\top \boldsymbol{\alpha}(\boldsymbol{\beta}^k))$$

and the elements of \mathbf{k}^k are

$$k_j^k = \text{sign}\left((\mathbf{X}^\top \nabla l(\mathbf{X}\boldsymbol{\beta}^k))_j + s_j\right) \cdot \min\left(\kappa, |(\mathbf{X}^\top \nabla l(\mathbf{X}\boldsymbol{\beta}^k))_j + s_j|\right),$$

for $j = 1, \dots, p$.

The proof is provided in Section SM 2.2 in the supplemental article [26].

The same approach can be used for the smoothed problem given in Equation 2.12. Using Equation 2.5 instead of Equation 2.4, the gap expression is now approximated by

$$\widetilde{\text{GAP}}_\mu(\boldsymbol{\beta}^k) \equiv f_\mu(\boldsymbol{\beta}^k) + l^*(\boldsymbol{\sigma}^k) + \psi_{\mu,k}^*(-\mathbf{X}\boldsymbol{\sigma}^k), \quad (3.9)$$

Where the Fenchel conjugate of Theorem 3 should be replaced by

$$\psi_{\mu,k}^*(\mathbf{v}) = \begin{cases} \frac{1}{2\lambda} \sum_{j=1}^p \left(\left[|v_j - \gamma(\mathbf{A}^\top \boldsymbol{\alpha}_\mu^*(\boldsymbol{\beta}^k))_j| - \kappa \right]_+^2 \right) + \frac{\gamma\mu}{2} \|\boldsymbol{\alpha}_\mu^*(\boldsymbol{\beta}^k)\|_2^2, & \text{if } \lambda > 0, \\ \frac{\gamma\mu}{2} \|\boldsymbol{\alpha}_\mu^*(\boldsymbol{\beta}^k)\|_2^2 & \text{if } \lambda = 0. \end{cases}$$

where $[\cdot]_+ = \max(0, \cdot)$.

Remark: It is worth noting that in spite of the massive use of \mathbf{X} and \mathbf{A} in this duality gap computation, it does not limit this approach and the obtained formula to the least square loss or the elastic net–total variation penalty. In fact, the very same expression holds for more general models with different \mathbf{X} , \mathbf{A} and l . It just requires that the loss function can be expressed in the form $l(\mathbf{X}\boldsymbol{\beta})$ with explicit l^* . Moreover, any constraint that fulfills the Nesterov assumption in Equation 2.4 could be used and eventually combined with the ℓ_1 or ℓ_2 norms as in Equation 2.2. For instance, by using results in Section D.2.3 of [32], one can easily adapt this for the logistic loss function.

3.4. Convergence analysis

In this section, we derive the sequences $(\mu^i)_{i=0}^\infty$ and $(\varepsilon^i)_{i=0}^\infty$ required for continuation steps of CONESTA. In each continuation step, i , FISTA (Algorithm 1) decreases f_μ^i until the error reaches ε^i . The resulting weight vector $\boldsymbol{\beta}_\mu^{i+1}$ will be the initial value for the next continuation step using updated parameters $(\mu^{i+1}, \varepsilon^{i+1})$.

The following theorem presents the main steps of CONESTA.

Theorem 4 (Convergence of CONESTA). *Let $(\mu^i)_{i=0}^\infty$ and $(\varepsilon^i)_{i=0}^\infty$ be defined recursively as*

$$\begin{cases} \varepsilon^0 = \tau \cdot \text{GAP}(\boldsymbol{\beta}^0), \\ \mu^0 = \mu_{\text{opt}}(\varepsilon^0), \\ \varepsilon_\mu^i = \varepsilon^i - \mu^i \gamma M, \\ \boldsymbol{\beta}_\mu^{i+1} = \text{FISTA}(\boldsymbol{\beta}_\mu^i, \mu^i, \varepsilon_\mu^i, \dots), \\ \varepsilon^{i+1} = \tau \cdot (\widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1}) + \mu^i \gamma M), \\ \mu^{i+1} = \mu_{\text{opt}}(\varepsilon^{i+1}), \end{cases} \quad (3.10)$$

where the vector $\boldsymbol{\beta}_\mu^i$ is the current value of $\boldsymbol{\beta}$ after the i th application of FISTA. The dual gap $\widetilde{\text{GAP}}_{\mu^i}(\boldsymbol{\beta}_\mu^{i+1})$ provides the current error for the smoothed and the original problem using Equation 2.10. Decreasing the error by the factor $\tau \in (0, 1)$ provides the current prescribed precisions for the original (ε^i) and for the smoothed problem (ε_μ^i) . μ_{opt} is defined by Equation 3.2.

Then, we have that

- (i) $\lim_{i \rightarrow \infty} \varepsilon^i = 0$, and
- (ii) $f(\boldsymbol{\beta}_\mu^i) \xrightarrow{i \rightarrow \infty} f(\boldsymbol{\beta}^*)$.
- (iii) If we require $f_{\mu_{\text{opt}}}(\boldsymbol{\beta}^k) - f(\boldsymbol{\beta}^*) < \varepsilon$ then the number of FISTA iterations is upper-bounded by

$$\frac{\sqrt{8\|\mathbf{A}\|_2^2 M \gamma^2 \|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}}{\varepsilon} + \frac{\sqrt{2L(\nabla(g))\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_2^2}}{\sqrt{\varepsilon}}.$$

- (iv) Suppose that the minimum β^* is unique, thereby, if we require $f(\beta_\mu^i) - f(\beta^*) < \varepsilon$, the total number of FISTA iterations is upper-bounded by $\mathcal{O}(C/\varepsilon)$, where $C > 0$ is a constant.

The proof of convergence is provided in Section SM 2.3 of the supplemental article [26].

Remarks Theorem 4 gives a rigorous proof that the continuation technique improves the convergence rate compared to the simple smoothing using a single value of μ . Indeed, it has been demonstrated in [5] (see also [15]) that the convergence rate obtained with single value of μ , even optimized, is $\mathcal{O}(1/\varepsilon) + \mathcal{O}(1/\sqrt{\varepsilon})$. However, the CONESTA algorithm achieves $\mathcal{O}(1/\varepsilon)$ for convex functions.

Theorem 4 uses the uniqueness hypothesis in (iv) just to get the boundedness of the FISTA iterations. Note that the authors in [11] have proved this property for a slightly modified version of the FISTA algorithm. We do not use these results for two reasons: First, this will slightly modify the μ_{opt} and other expressions which we have not include in our simulations. Second, this result must be adapted to the smoothed version. In fact, we will need to establish the boundedness of the FISTA iterated uniformly with respect to μ as μ converges to zeros. On the other hand, if the objective function is strongly convex with strong convexity parameter $\sigma > 0$, then the FISTA algorithm enjoy global linear convergence rate and especially ensures the convergence of β^k to the minimum β^* . In fact, we have that $\|\beta^k - \beta^*\|_2 < \sqrt{2\sigma/\varepsilon}$ as far as $f(\beta^k) - f(\beta^*) < \varepsilon$ (see remark 3 in [15]). In this case, not only the hypothesis in (iv) is naturally satisfied, but also it is a favorable case to apply a continuation technique since the estimations on the successive intermediate β_μ^i will ensure a faster convergence contrary to the simple convergence case in which we don't make use of this rate. Nevertheless, we did not include this case in this paper since it will engender the modification of the optimal smoothing value μ_{opt} and consequently all the other estimations due to the different convergence rate of FISTA. All these questions will be considered in future works.

3.5. The algorithm

Most inputs of CONESTA Algorithm 2 are already detailed above for FISTA (Algorithm 1). Note that all functions that depend on μ in FISTA assume that this is a fixed value, while it is a free input parameter of CONESTA. In fact, it is a parameter that is calculated by CONESTA. The latter admit additional parameters: the duality gap function, $\widetilde{\text{GAP}}_\mu(\cdot)$, defined in Equation 3.9, the factor τ , defined in Theorem 4, and the scalar M , defined in Equation 2.10.

Here we explain some of the steps of the algorithm and give references to the equations and theorems that are used.

- In Line 1, the duality gap is used to estimate the current error, $f(\beta^0) - f(\beta^*)$, at the start vector β^0 for the non-smooth problem. In this particular case, we use $\widetilde{\text{GAP}}_\mu(\cdot)$ with a negligible smoothing value of *e.g.* $\mu = 10^{-8}$.

- Line 2 selects the optimal smoothing parameter, μ^0 , such that we reach the prescribed precision ε^0 , when using FISTA.
- Line 5 defines the stopping precision for the FISTA loop. This choice ensures that at the next step, *i.e.* on Line 6, $\widetilde{\text{GAP}}_{\mu^i}(\beta^{i+1}) < \varepsilon^i - \mu^i \gamma M$ which implies that the error will be smaller than ε^i . Note that by selecting the maximum compared to ε , we avoid situations in which FISTA converges to precisions smaller than the globally prescribed one.
- Line 7 estimates the current error (after FISTA) toward the non-smooth problem using $\widetilde{\text{GAP}}_{\mu^i}(\cdot) + \mu^i \gamma M$. We prefer to use $\widetilde{\text{GAP}}_{\mu}(\beta)$ over $\widetilde{\text{GAP}}(\beta)$ because it converges to zero for any fixed μ unlike $\widetilde{\text{GAP}}(\beta)$. Thus the application of FISTA will converge as shown in the proof of Theorem 4.
- Line 8 checks the stopping criterion for the main algorithm. Together with Line 5 it ensures that the final computed β^i satisfies $f(\beta^i) - f(\beta^*) < \varepsilon$. Roughly speaking, this is equivalent to inspecting the final stopping criterion, $f(\beta^i) - f(\beta^*) < \varepsilon$, at every iteration of the inner FISTA loop.
- Line 11 computes the next precision ε^{i+1} in a decreasing geometric progression governed by τ .
- Line 12 updates the smoothing parameter by selecting the optimal one with respect to the precision ε^{i+1} .

4. Application to linear regression with elastic net and total variation penalties

This section describes how to implement the theory described above for the minimization problem of interest in this paper, namely that in Equation 2.2. The input parameters of the CONESTA algorithm were: a random starting vector $\beta^0 \in \mathbb{R}^p$; a precision $\varepsilon = 10^{-6}$; $M = \max_{\alpha \in \mathcal{K}} \frac{1}{2} \|\alpha\|_2^2 = p/2$. For τ ,

Algorithm 2 CONESTA

Input: $\beta^0, \varepsilon, \tau, \mathbf{A}, \gamma, \nabla(g + \gamma s_{\mu})(\cdot), \text{prox}_{t_{\mu} \kappa h}(\cdot), L(\nabla(g)), \widetilde{\text{GAP}}_{\mu}(\cdot), M, \beta^0 \in \mathbb{R}^p, \tau \in (0, 1)$
Output: β^i such that $f(\beta^i) - f(\beta^*) < \varepsilon$

```

1:  $\varepsilon^0 = \tau \cdot \widetilde{\text{GAP}}_{\mu=10^{-8}}(\beta^0)$ 
2:  $\mu^0 = \mu_{opt}(\varepsilon^0)$ 
3:  $i = 0$ 
4: loop
5:    $\varepsilon_{\mu}^i = \max(\varepsilon, \varepsilon^i - \mu^i \gamma M)$ 
6:    $\beta^{i+1} = \text{FISTA}(\beta^i, \mu^i, \varepsilon_{\mu}^i, \mathbf{A}, \gamma, \nabla(g + \gamma s_{\mu^i})(\cdot), \text{prox}_{t_{\mu^i} \kappa h}(\cdot), L(\nabla(g)), \widetilde{\text{GAP}}_{\mu^i}(\cdot))$ 
7:    $\varepsilon^i = \widetilde{\text{GAP}}_{\mu^i}(\beta^{i+1}) + \mu^i \gamma M$ 
8:   if  $\varepsilon^i \leq \varepsilon$  then
9:     break
10:  end if
11:   $\varepsilon^{i+1} = \tau \cdot \varepsilon^i$ 
12:   $\mu^{i+1} = \mu_{opt}(\varepsilon^{i+1})$ 
13:   $i \leftarrow i + 1$ 
14: end loop
```

we tested three different values, namely $\tau = 0.8$, $\tau = 0.5$ and $\tau = 0.2$, and found very similar performances. We therefore chose to use $\tau = 0.5$ in all of what follows. The Lipschitz constant of the gradient of least-squares loss with ℓ_2 penalty is computed as $L(\nabla(g)) = \lambda_{\max}(\mathbf{X}^\top \mathbf{X}) + \lambda$, where $\lambda_{\max}(\mathbf{X}^\top \mathbf{X})$ is the largest eigenvalue of $\mathbf{X}^\top \mathbf{X}$.

To apply the presented theory in the specific context with a TV penalty, we need to define the linear operator for Nesterov's smoothing (denoted \mathbf{A}) and how to compute the dual variables. The former is used to calculate the gradient $\nabla(g + \gamma s_\mu)$ (Equation 2.7) and the gap $\widetilde{\text{GAP}}_\mu(\cdot)$ (Equation 3.9).

Now we show how we can express the TV operator in a dual form on β , i.e. in the form

$$TV(\beta) = \max_{\alpha_{TV} \in \mathcal{K}_{TV}} \langle \alpha_{TV} | \mathbf{A}_{TV} \beta \rangle, \quad (4.1)$$

where \mathcal{K}_{TV} is a compact space and \mathbf{A}_{TV} is a linear operator. This formulation, introduced in Equation 2.4, is at the core of CONESTA. It allows the determination of the linear operator \mathbf{A} and the computation of α_μ^* .

We consider the general case where we have n input 3D images that are not defined over the entire $p_x \times p_y \times p_z$ grid but over a mask $\mathcal{M} \subset \{(i, j, k) \in \mathbb{Z}^3 \mid 1 \leq i \leq p_x, 1 \leq j \leq p_y, 1 \leq k \leq p_z\}$ such that $|\mathcal{M}| = p$ with $p \leq p_x \cdot p_y \cdot p_z$. Let $\phi : \mathcal{M} \rightarrow \{1, 2, \dots, p\}$ be the function that maps 3D coordinates, within the mask, onto the one-dimensional index of the masked and collapsed image. Finally, we denote by $\text{grad}_{(i,j,k)}(\mathbf{x})$ the spatial gradient of the collapsed vector \mathbf{x} at $\phi(i, j, k)$. We have, by definition, that

$$TV(\beta) \equiv \sum_{(i,j,k) \in \mathcal{M}} \|\text{grad}_{(i,j,k)}(\beta)\|_2. \quad (4.2)$$

Let (i, j, k) be a voxel inside the mask. In the context of TV, a group of variables G (as in Equation 2.3) is composed of this voxel and three of its “face neighbors”, when they exist, namely voxels $(i+1, j, k)$, $(i, j+1, k)$ and $(i, j, k+1)$. Let us consider first the case where all the neighbors of (i, j, k) also belong to the mask:

$$\text{grad}_{(i,j,k)}(\beta) = \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_G^r} \underbrace{\begin{bmatrix} \beta_{\phi(i,j,k)} \\ \beta_{\phi(i+1,j,k)} \\ \beta_{\phi(i,j+1,k)} \\ \beta_{\phi(i,j,k+1)} \end{bmatrix}}_{\beta_G}.$$

The \mathbf{A}_G^r is a linear operator from the reduced space \mathbb{R}^4 and can be extended to \mathbb{R}^p in order to be directly applied on β . We denote this extended version \mathbf{A}_G . The other cases, where some neighbors of the considered point are outside the mask, are not detailed here, but do not change the theoretical result. Using Equation 2.3, we obtain a dual form of the TV:

$$\begin{aligned} TV(\beta) &= \sum_{G \in \mathcal{G}} \|\mathbf{A}_G \beta\|_2 \\ &= \sum_{G \in \mathcal{G}} \max_{\alpha_G \in \mathcal{K}_G} \langle \alpha_G | \mathbf{A}_G \beta \rangle. \end{aligned}$$

As a consequence of this dual formulation, we obtain the desired result, namely that

$$TV(\beta) = \max_{\alpha_{TV} \in \mathcal{K}_{TV}} \langle \alpha_{TV} | \mathbf{A}_{TV} \beta \rangle,$$

where $\mathcal{K}_G = \{\alpha_G \in \mathbb{R}^3, \|\alpha_G\|_2 \leq 1\}$. The dual variable α_{TV} is the concatenation of all α_G and belongs to $\mathcal{K}_{TV} = \Pi_{G \in \mathcal{G}} \mathcal{K}_G$. The \mathbf{A}_{TV} is the vertical concatenation of all the \mathbf{A}_G matrices, and \mathcal{K}_{TV} , being the product of all the compact spaces \mathcal{K}_G , is itself a compact space.

To calculate $\widehat{\text{GAP}}_\mu(\cdot)$ and the gradient $\nabla(g + \gamma s_\mu)$ we need to define the projection onto \mathcal{K} . The latter is necessary in order to compute $\alpha_\mu^*(\beta)$ (see Equation 2.11) which is itself a function of the projections onto each compact set \mathcal{K}_G , defined by

$$\text{proj}_{\mathcal{K}_G}(\mathbf{x}) = \frac{\mathbf{x}}{\max(\|\mathbf{x}\|_2, 1)} = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\|_2 \leq 1 \\ \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & \text{otherwise.} \end{cases} \quad (4.3)$$

The projection onto \mathcal{K} is then the concatenation of all the projections onto the sets \mathcal{K}_G .

5. Experiments on simulated 1D data set where the solution of $\min_\beta f$ is known

In this section we propose an extensive comparison of CONESTA with other state-of-the-art algorithms on simulated data, where we control all aspect of the data: the true minimizer, β^* , associated with the Lagrange multipliers κ , λ and γ .

5.1. Data set generation

The experimental setup for the simulated 1D data set was inspired by that of Bach et al. [3] and is shown in Table 1. This setup is a designed experiment with multiple small- and medium-scale data sets having combinations of low, medium and high degrees of correlation between variables, low, medium and high levels of sparsity, and low, medium and high signal-to-noise ratios.

A candidate version of the predictors, *i.e.* the \mathbf{X}_0 matrix, was drawn from a Gaussian distribution $\mathcal{N}(\mathbf{1}, \Sigma)$, where Σ was generated according to the constant correlation model described by Hardin, Ramon Garcia and Golan [27], where $\text{diag}(\Sigma) = 1$ and off-diagonal elements are all equal to ρ . The ρ is drawn from a distribution that approximately follows the Gaussian distribution $\mathcal{N}(0, d_c/\sqrt{n})$. Here, d_c is the parameter controlling the dispersion of the correlation, and takes values of $d_c = 1$ in the low-correlation case, $d_c = 4.5$ in the mid-correlation case and $d_c = 8$ in the high-correlation case.

Concerning the weights vector, β^* , a sparsity parameter fixes the percent of null weights within it. The remaining weights, that contribute to the prediction, were drawn from $\mathcal{U}(0, 1)$ and sorted in ascending order.

The elements of the error vector, \mathbf{e} , were all drawn from a Gaussian distribution, $\mathcal{N}(1, 1)$, and then scaled to unit ℓ_2 -norm.

Given a candidate data matrix \mathbf{X}_0 , the true minimizer β^* , the true residual vector \mathbf{e} , and regularization constants κ , λ and γ , that were arbitrarily set to $\kappa = 0.618$, $\lambda = 1 - \kappa$ and $\gamma = 1.618$, we generated the simulated data set (\mathbf{X}, \mathbf{y}) such that β^* is the solution that minimizes Equation 2.2, with $s(\beta^*) = TV(\beta^*)$. The data matrix \mathbf{X} is obtained by finding scaling factors to be applied on the columns of the candidate data matrix \mathbf{X}_0 . All the details of the scaling procedure to produce a data set with a known exact solution is detailed in the ancillary paper [31]. Once \mathbf{X} is found, the \mathbf{y} vector is computed as $\mathbf{y} = \mathbf{X}\beta^* - \mathbf{e}$.

The main benefit of generating the data in this way is that we know everything about it. In particular, we know the true minimizer, β^* , and we know the Lagrange multipliers κ , λ and γ . There is no need to use *e.g.* cross-validation to find any parameters, and we can immediately determine whether the β^k we obtain is close to the true β^* or not.

Five runs were performed for each of these settings. A total of 405 simulation runs were thus performed. For each combination of settings, we generate data such that we knew the exact solution of the given minimization problem.

5.2. Result of the comparison

We applied the minimization algorithms to simulated data that were generated using the settings shown in Table 1. For each of the settings of the designed experiment, we measured the number of iterations and the time (in seconds) required to reach a certain precision level, ε , with each of the tested minimization algorithms. For each precision level, ranging from 1 to 10^{-6} , for each algorithm and for each data set, we *ranked* each algorithm according to the time it consumed in order to reach the given precision. Then, we averaged the ranks across data sets in order to evaluate the significance of the difference in ranks between algorithms. We used the Friedman test [22] to do this.

As seen in Figure 1 and Table 2, the excessive gap method is the fastest algorithm for the lowest (*i.e.*, largest) precision, $\varepsilon = 1$. It significantly outperformed CONESTA; ADMM was the second fastest algorithm, but it was not significantly faster than CONESTA. CONESTA was the third fastest, it significantly

TABLE 1

The experimental setup for the first simulation study. The parameters varied were: the size of the data set (n, p), the correlation between variables, the sparsity of the data, and the signal-to-noise ratio. Five runs were performed for each combination of settings in order to assess the variability of the different algorithms. The medium level for sparsity (marked with an asterisk, ‘’) was only performed for the low and medium sizes.*

| Level | Size ($n \times p$) | Correlation | Sparsity | Signal-to-noise ratio |
|--------|-----------------------|-------------|----------|-----------------------|
| Low | 200×200 | Low | 50 % | 0.5 |
| Medium | 632×1514 | Medium | 72.5 %* | 1.0 |
| High | 2000×10000 | High | 95 % | 5.0 |

TABLE 2

Average rank of the convergence speed of the algorithms to reach precisions $(f(\beta^k) - f(\beta^*))$ ranging from 1 to 10^{-6} . We have here reported whether the average rank of a given algorithm was significantly larger $>$ (slower) or significantly smaller $<$ (faster) than CONESTA (a missing ' $>$ ' or ' $<$ ' means non-significant, i.e. $p > 0.05$). We performed a post hoc analysis using the Friedman test. All reported significances were $p < 10^{-3}$ or smaller, corrected for multiple comparisons. For a given data set, all algorithms were evaluated with a limited upper execution time. Thus, some high precisions (e.g., $10^{-5}, 10^{-6}$) were not always reached within the limited time. For some algorithms (FISTA with medium/large μ), the high precisions may in fact not be reachable at all. In those situations, the execution time was set to $+\infty$.

| Algorithm | Average rank of the speed to reach a given precision | | | | | | |
|--------------------|--|------------|------------|--------------|--------------|--------------|--------------|
| | 1 | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 10^{-6} |
| CONESTA | 3.3 – | 2.7 – | 2.2 – | 1.6 – | 1.3 – | 1.0 – | 1.0 – |
| ADMM | 2.9 | 2.1 | 1.9 | 1.8 | 1.7 $>$ | 1.8 $>$ | 1.5 $>$ |
| Excessive gap | 1.8 $<$ | 2.2 | 2.0 | 2.3 $>$ | 2.3 $>$ | 2.2 $>$ | 1.7 $>$ |
| FISTA medium μ | 4.7 $>$ | 6.0 $>$ | 4.6 $>$ | 3.4 $>$ | 2.7 $>$ | 2.2 $>$ | 1.7 $>$ |
| FISTA large μ | 2.9 | 3.2 | 3.7 $>$ | 3.4 $>$ | 2.7 $>$ | 2.2 $>$ | 1.7 $>$ |
| FISTA Chen's μ | 6.2 $>$ | 5.0 $>$ | 4.4 $>$ | 3.4 $>$ | 2.7 $>$ | 2.2 $>$ | 1.7 $>$ |
| Inexact FISTA | 6.2 $>$ | 5.4 $>$ | 4.3 $>$ | 3.3 $>$ | 2.7 $>$ | 2.2 $>$ | 1.7 $>$ |

outperformed the inexact FISTA and FISTA with medium or small μ .

The average convergence rank of CONESTA improved for the higher precision levels. CONESTA remained the third fastest algorithm, for levels $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-2}$, after ADMM and the excessive gap method, but it was no longer significantly slower than ADMM. However superiority over all FISTA with Nesterov's smoothing and inexact FISTA remained, and became significant.

For all precisions higher (i.e., smaller) than 10^{-3} , CONESTA outperformed all other algorithms. The superiority of CONESTA is significant for all precisions, except for the comparison with ADDM at $\varepsilon = 10^{-3}$.

We did not perform any particular code optimization in order to fairly measure the time difference between the algorithms. Moreover, many solvers share large portions of codes, especially all FISTAs, Inexact FISTA and CONESTA. However, to avoid any suspicion of bias due to our implementation, Figure 2 presents error as a function of the number of iterations.

ADMM appears to be particularly fast in the beginning, where it converges quickly to the middle precision levels, after which it starts to level and be outperformed by CONESTA. This ADMM behavior is well-known, and is discussed by e.g. Boyd et al. [9].

The inexact FISTA performs impressively when only looking at the number of iterations. However on large data sets, it took a similar number of iterations as CONESTA to reach the same precision. This implies an advantage to CONESTA, namely that it should scale better with larger data, i.e. that its asymptotic time complexity be lower than that of ADMM.

6. Experiments on a structural MRI data set

6.1. Structural MRI experimental data set

The data used in the preparation of this article were obtained from the database of the Alzheimer's disease (AD) neuroimaging initiative (ADNI) (<http://adni.loni.usc.edu/>).

The MRI data set included standard T1-weighted images obtained with different 1.5-T scanner types using a three-dimensional MP-RAGE sequence or equivalent protocols with varying resolutions. Images were post-processed to correct for some artifacts [29]. As a result, 509 images [17] were segmented into GM (Gray Matter), WM (White Matter) and CSF (Cerebrospinal Fluid) using

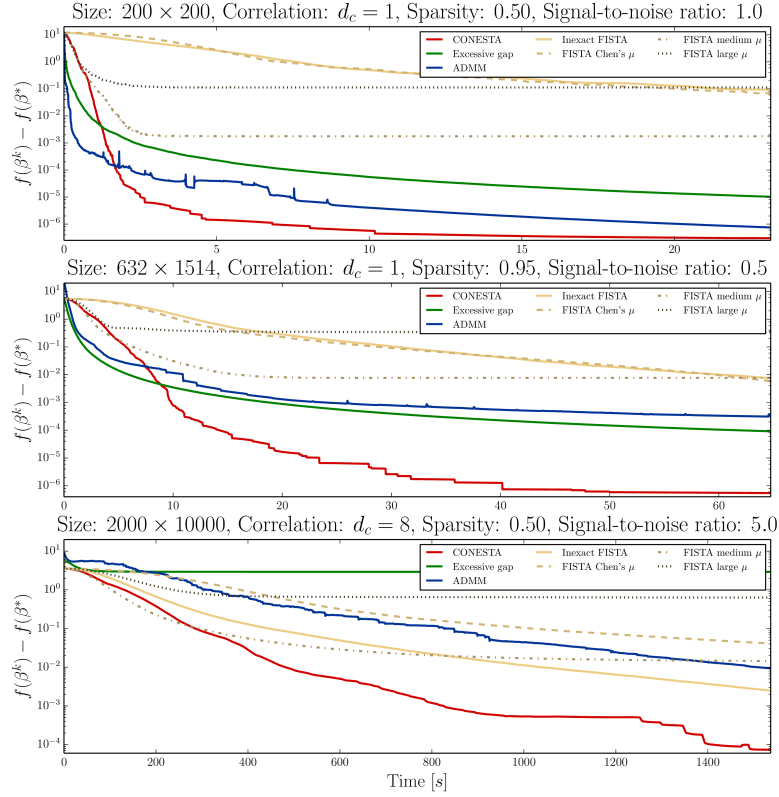


Fig 1: The error as a function of the computational time. The vertical axis is plotted on a logarithmic scale. The settings used are given in the figure titles.

the SPM8 unified segmentation routine [2].

A total of 456 images were retained after quality control on GM probability. These images were spatially normalized into a template (dimension: $121 \times 145 \times 121$, voxels size: 1.5 mm isotropic) using DARTEL [1] and modulated with the Jacobian determinants of the nonlinear deformation field.

We retain GM voxels with a minimum value of 0.01 and at least 10^{-6} of standard-deviation across the participants' cohort. Then each voxel was centered and scaled at the cohort level. Those masked, warped and modulated GM images (286 214 voxels) were completed with three demographic predictors (age, gender and education level) leading to $p = 286\,217$ input features. Those three demographic predictors were excluded from any penalization.

Since the ultimate goal of such machine learning approaches is to predict the

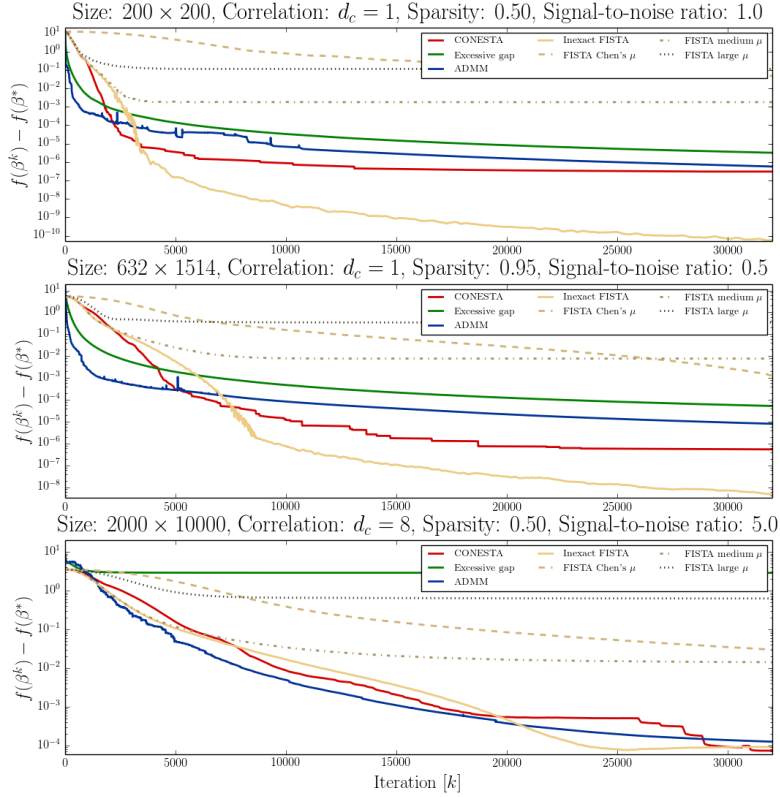


Fig 2: The error as a function of the number of iterations. The vertical axis is plotted on a logarithmic scale. The settings used are given in the figure titles.

future “medical” outcome of subjects, we used as the target variable (\mathbf{y}), the ADAS (Alzheimer’s Disease Assessment Scale-Cognitive Subscale) score measured 800 days after the acquisition of the brain images. The ADAS score is one of the most frequently used tests to measure cognition in clinical trials and it is provided in the ADNI data set.

As participants, we considered one group of 119 control subjects (CTL) that never converted to AD within the six years of the study. As patients, we considered one group of patients with mild cognitive impairment (MCI) that converted to AD within 800 days. We pooled those two groups leading to a data set with $n = 199$ subjects.

6.2. Relevance of the TV penalty in the context of neuroimaging

Before comparing convergence speed of the optimization algorithms, this section demonstrates the improvements obtained with structured sparsity; or more precisely with TV in the context of neuroimaging. Indeed, the ultimate purpose of CONESTA is to provide an efficient algorithm to extract relevant biomarkers of brain diseases thanks to the benefit of the TV penalty in the context of very high dimensional data ($> 10^5$ features). Consequently, this section compares performances with and without TV, independent of the optimization algorithms.

Models were compared using 5-fold cross-validation (5CV). To assess the predictive power of the model, we computed the coefficient of determination, R -squared (R^2), obtained on the left-out samples.

Along with the expectation of improved prediction performances, the TV penalty targets a more important goal: the estimation of reproducible β maps against variations of the learning samples. Indeed, clinicians expect that the identified biomarkers, *i.e.* the non-null weights of the β map, to be similar if other patients, with similar clinical conditions, would have been used.

We used two similarity measures to assess the stability of those β maps across the re-sampling. First we computed the mean correlation between pairs of β maps obtained across the 5CV: pairwise correlations were transformed into z -scores using Fisher’s z -transformation, averaged and finally transformed back into an average correlation denoted \bar{r}_β . Second, we computed an inter-rater agreement measure: the Fleiss’ κ statistics, to assess the agreement between supports (non-null weights of β maps) recovered by the different penalties across the 5CV.

The weights of the five β maps were partitioned into 3 categories according to their signs. This led to negative, positive and, out-of-support voxels. The Fleiss’ kappa statistics, κ_β , was then computed for the five raters. Since the 5CV folds share many ($\sim 64\%$) of their training samples, no unbiased significance measure can be directly obtained from the \bar{r}_β and κ_β statistics. Thus, we used permutation testing to assess empirical p -values. The null hypothesis was simulated by 1 000 random permutations of the target variable, \mathbf{y} , within each permuted sample we executed the whole 5CV round. Then the statistics on the true data were compared with the ones obtained on the permuted ones.

TABLE 3
 Predictive performances: R -squared (R^2) and stability of β maps measured as the average correlation between pairs of \bar{r}_β (using Fisher's z -transformation) and the inter-rater measure (κ_β , Fleiss' κ statistic) of agreement between the supports recovered across the 5CV. Empirical p -values were assessed by 1 000 executions of the 5CV round on randomly permuted data. Significance notations: ***, $P \leq 10^{-3}$, **, $P \leq 10^{-2}$, *, $P \leq 0.05$.

| Method | Prediction | | Stability of β maps | | | |
|------------------------|------------|----------|---------------------------|----------|----------------|----------|
| | R^2 | Δ | \bar{r}_β | Δ | κ_β | Δ |
| ℓ_1 | 0.50* | +0.02** | 0.58*** | +0.09*** | 0.29*** | +0.07*** |
| $\ell_1 + TV$ | 0.52** | | 0.67*** | | 0.36*** | |
| $\ell_2 + \ell_1$ | 0.53* | -0.02 | 0.58*** | +0.07*** | 0.33*** | +0.02*** |
| $\ell_2 + \ell_1 + TV$ | 0.51** | | 0.65*** | | 0.35*** | |

The predictive performances, displayed in Table 3, were not clearly improved with the TV penalty. However, $\ell_1 + TV$ was found significantly better than ℓ_1 alone.

Figure 3 demonstrates that the TV penalty provides a major breakthrough in terms of support recovery of the predictive brain regions. Contrary to the elastic net ($\ell_1 + \ell_2$) penalty, that highlights scattered and meaningless patterns, the elastic net + TV (*i.e.*, $\ell_1 + \ell_2 + TV$) penalties provide a smooth map that match the well-known brain regions involved in AD [23]. The most important cluster of non-null weights was found in the left hippocampus and the ventricular enlargement was identified. The right panel of Figure 3 simply confirm that TV is producing much more stable supports than elastic net only. Without TV, some (few) voxels within the hippocampus were at most selected 3 times during the 5CV round. Such results demonstrate that elastic net is effectively useless in the context of biomarker identification. Conversely, with the TV penalty, voxels within the hippocampus were always selected. The measures of the stability of β maps presented in Table 3 demonstrated that TV significantly improves the reproducibility of the predictors map leading to meaningful biomarkers.

6.3. Comparisons of the convergence speed of the minimization algorithms

After having demonstrated the relevance of TV to extract stable biomarkers, we go back to the comparison of the convergence speed of CONESTA to that of the other state-of-the-art minimization algorithms mentioned in Section 2.2, namely: the excessive gap method, FISTA with three different fixed smoothing parameters (μ , see Section 2.4) and inexact FISTA (the proximal operator is approximated).

Note: ADMM has been excluded in this example. The version by Wahlberg et al. [48] is made for 1D underlying data, and the authors did not adapt the method to the case of 2D or 3D underlying data. Note also that without the particular form of the $A^\top A$ matrix (it is tridiagonal in the $\ell_1 + TV_{1D}$ case), ADMM would perform much slower, because it requires us to solve a linear

system in each iteration. This is thus a major drawback of ADMM, that it can not easily be adapted to arbitrary complex penalties.

For FISTA with different values of μ , we chose the three values of μ as follows: (i) Chen's μ with $\mu = \frac{\epsilon}{2\gamma M}$ as explained in Section 2.4 (used by Chen et al. [15]), (ii) Medium $\mu = (\text{Chen's } \mu)^{1/2}$ and (iii) Large $\mu = (\text{Chen's } \mu)^{1/4}$. The first proposal for μ ensures that we reach the desired precision, but it may be slow to converge. The two other proposals are larger values of μ that do not guarantee that we reach the desired precision before convergence. But as we will see, FISTA with different values of μ will be tough competitors in the beginning when evaluating whether CONESTA can match their convergence speed.

First we fixed a desired precision, $\epsilon = 10^{-6}$, which was used as the stopping criterion for all algorithms. It was also used to derive the smoothing parameter for FISTA with fixed μ .

TABLE 4
Execution time ratios of each state-of-the-art algorithm over the time required by CONESTA to reach the same precision.

| Algorithm | Time ratio over CONESTA to reach a given precision | | | | | | |
|------------------------|--|--------|-----------|-----------|-----------|-----------|-----------|
| | 10^1 | 10^0 | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} |
| Excessive gap | 0.97 | 0.39 | 0.24 | 1.05 | 1.08 | 1.74 | 3.64 |
| FISTA Chen's μ | 2.7 | 10.65 | 17.5 | 29.95 | 16.72 | 13.47 | 10.45 |
| FISTA large μ | 1.04 | 0.97 | 1.12 | 1.14 | — | — | — |
| FISTA very large μ | 0.99 | 0.97 | 1.32 | — | — | — | — |
| Inexact FISTA | 1.13 | 1.92 | 2.64 | 3.97 | 4.38 | 5.45 | 6.07 |

Figure 4 shows that FISTA with fixed μ is either too slow (Chen's μ) or, as

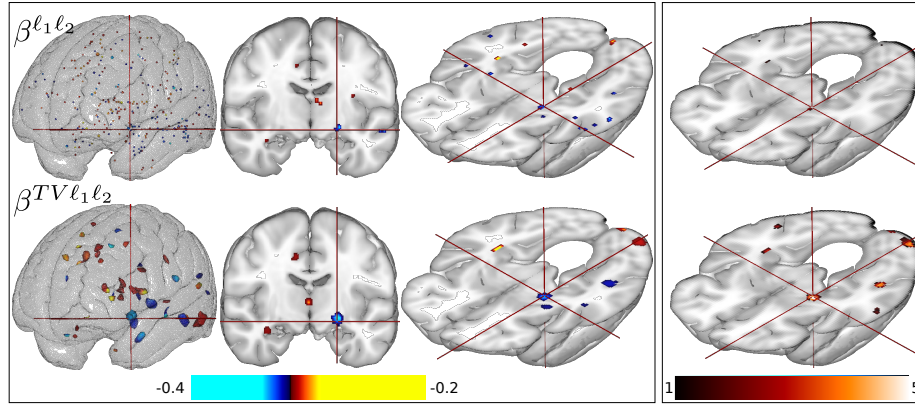


Fig 3: Left panel: weights map found with elastic net ($\beta^{\ell_1 \ell_2}$) and elastic net plus TV penalties ($\beta^{\ell_1 \ell_2 TV}$) found using CONESTA. Right panel: count the number of time voxels were part of the support (non-null weights) across the 5CV.

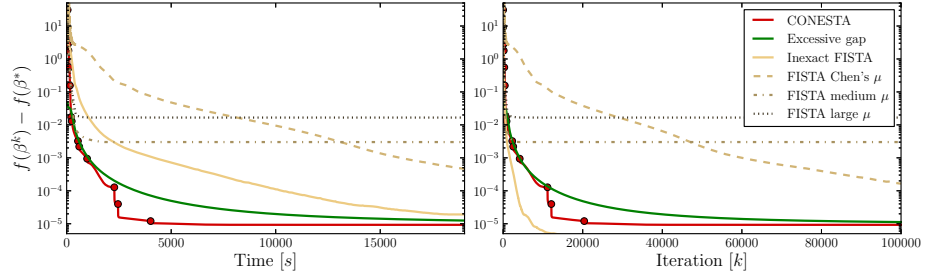


Fig 4: The error as a function of the computational time (left plot) and the number of iterations (right plot). The vertical axis is plotted on a logarithmic scale. Dots on the CONESTA curve indicate where the continuation steps take place, *i.e.* where the dynamic selection of a new smoothing parameter happened.

expected, does not reach the desired precision (as with the two large μ). However, CONESTA compete with FISTA with large μ and excessive gap, during the first iterations. This demonstrates two points: CONESTA dynamically picked an efficient (large enough) smoothing parameter and that the gap stopping criterion, used in the nested FISTA loop, allows us to stop before reaching a plateau and chose a smaller μ .

Figure 4, right panel, illustrates that the number of iterations of inexact FISTA is low. However, the left panel of Figure 4, shows that it is always considerably slower, in terms of the execution time, compared to the excessive gap method or CONESTA.

Table 4 reveals that the execution time (as the ratio over CONESTA's execution time), to reach a given precision, increases up to a factor six. This demonstrates the hypothesis we stated in Section 2.2 that inexact FISTA becomes slower after many iteration due to the necessity to decrease the precision as $1/k^4$ (k being the number of FISTA iterations), in the approximation. We experimentally observed the following phenomenon: during the first 100 FISTA iterations, only one iteration of the nested loop was necessary, but then this number increase progressively and exceeded 100 after few thousand FISTA iterations.

As a conclusion, Figure 4 and Table 4 demonstrate that on high-dimensional data sets, CONESTA outperformed all other algorithms, except in the very first iterations where excessive gap was faster, after which it is largely overtaken by CONESTA.

7. Conclusion

We investigated in this paper the optimization problem where a linear regression model was combined with several convex penalties, including a structured penalty under very general assumptions on the structured penalty and the smooth loss function.

We proposed an algorithm, called CONESTA, that aims to overcome a set of difficulties experienced with current state-of-the-art algorithms, like the non-separability and non-smoothness of the penalty. Our results show that the proposed algorithm enjoys both desirable theoretical convergence guarantees and practical scalability properties under various difficult settings involving complex structured constraints, and high-dimensionality. Specifically, the main benefits over the state-of-the-art methods can be summarized in the following points:

1. CONESTA enjoys a converge rate of $\mathcal{O}(1/\varepsilon)$ which is an improvement over proximal gradient smoothing methods (with fixed smoothing) whose rate is $\mathcal{O}(1/\varepsilon) + \mathcal{O}(1/\sqrt{\varepsilon})$, [5, 15].
2. On simulated and experimental data CONESTA outperformed (in terms of execution time and precision of the solution) several state-of-the-art optimization algorithms, such as the excessive gap method, ADMM and an *inexact* proximal gradient algorithm.
3. CONESTA has been designed to be a robust solver that alleviates practical problems affecting other optimizers in very high dimension: (i) Excessive gap method does not allow true sparsity nor complex loss function. (ii) *Inexact* proximal gradient algorithm, which is the most commonly used algorithm, has a better convergence rate (in term of proximal iterations) than CONESTA. However, as observed on very high dimensional MRI dataset after hundreds of iterations, solving the subproblem, using an inner FISTA loop, makes it much slower (4 times to reach $\varepsilon < 10^{-3}$) than CONESTA. (iii) ADMM is often not robust to the exact choice of the parameters. Conversely CONESTA requires only a global precision and, thanks to the duality gap, it will dynamically adapt the decreasing smoothing sequence.
4. The algorithm that we propose is able to include any combination of terms from a variety of smooth and non-smooth penalties. The algorithm does not require the proximal operators, or any other auxiliary minimization problem, contrary to many other similar state-of-the-art algorithms. For instance, any penalty in the form of a sum of q -norms of linear operators applied on all or on a subset of all variables could be used with this algorithm together with any smooth loss function for which we know the gradient.
5. Thanks to the proposed duality gap, CONESTA has a rigorous stopping criterion that allows a desired precision to be reached with theoretical certainty.

Finally CONSETA has been designed to a good candidate for warm restart. Indeed the whole smoothing sequence is dynamically adapted to the distance of the global minimum. This is a crucial quality for a solver in real life conditions where the user will explore a grid of possible penalty parameters. Since a first solution obtained with a set of parameter is similar to that sought with a slight modification of one parameter, the first solution will be a efficient starting point to accelerate the resolving of the second problem.

Appendix A: Supplementary Material

Supplementary document.

- *Title:* Supplement to “An Iterative Smoothing Algorithm for Regression with Structured Sparsity”
- *Url:* <https://hal-cea.archives-ouvertes.fr/cea-01324021>
- *Description:* This document contains proofs and supplementary details for the current paper. All section numbers and equation numbers in this supplementary document are preceded by the letters A, to distinguish them from those from the current paper.

Supplementary materials 1/2: The ParsimonY Python library.

- *Url:* <https://github.com/neurospin/pylearn-parsimony>
- *Description:* ParsimonY is Python library for structured and sparse machine learning. ParsimonY is open-source (BSD License) and compliant with scikit-learn API.

Supplementary materials 2/2: data sets and scripts.

- *Url:* ftp://ftp.cea.fr/pub/unati/brainomics/papers/ols_nestv
- *Description:* This url provides the data sets and the Python scripts used for the paper. We also provide a simple example that illustrates the usage of the ParsimonY library on neuroimaging data set.

Acknowledgments

We would like to acknowledge Denis Rivière and Jean-François Mangin from NeuroSpin, CEA, Paris-Saclay, Gif-sur-Yvette - France. We also would like to acknowledge Elvis Dohmatob from PARIETAL - Inria Saclay - France.

Data collection and sharing for this project was funded by the Alzheimer’s Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer’s Association; Alzheimer’s Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support

ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer’s Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

- [1] ASHBURNER, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage* **38** 95–113.
- [2] ASHBURNER, J. and FRISTON, K. J. (2005). Unified segmentation. *NeuroImage* **26** 839–851.
- [3] BACH, F., JENATTON, R., MAIRAL, J. and OBOZINSKI, G. (2011). Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning* (S. Sra, S. Nowozin and S. J. Wright, eds.) MIT Press.
- [4] BECK, A. and TEOULLE, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* **2** 183–202.
- [5] BECK, A. and TEOULLE, M. (2012). Smoothing and First Order Methods: A Unified Framework. *SIAM Journal on Optimization* **22** 557–580.
- [6] BECKER, S., BOBIN, J. and CANDÈS, E. J. (2011). NESTA: A Fast and Accurate First-Order Method for Sparse Recovery. *SIAM Journal on Imaging Sciences*.
- [7] BONNANS, J. F., GILBERT, J. C. and LEMARECHAL, C. (2006). *Numerical Optimization: Theoretical and Practical Aspects*, 2nd ed. Springer-Verlag Berlin and Heidelberg GmbH & Co. K.
- [8] BORWEIN, J. M. and LEWIS, A. S. (2006). *Convex Analysis and Non-linear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer.
- [9] BOYD, S., PARIKH, N., CHU, E., PELEATO, B. and ECKSTEIN, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* **3** 1–122.
- [10] CHAMBOLLE, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision* **20** 89–97.
- [11] CHAMBOLLE, A. and DOSSAL, C. (2015). On the Convergence of the Iterates of the “Fast Iterative Shrinkage/Thresholding Algorithm”. *Journal of Optimization Theory and Applications* **166** 968–982.
- [12] CHAMBOLLE, A. and POCK, T. (2010). A first-order primal-dual algorithm for convex problems with applications to imaging.
- [13] CHAMBOLLE, A. and POCK, T. (2011). A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.* **40** 120–145.

- [14] CHEN, X. and LIU, H. (2011). An Efficient Optimization Algorithm for Structured Sparse CCA, with Applications to eQTL Mapping. *Statistics in Biosciences* **4** 3–26.
- [15] CHEN, X., LIN, Q., KIM, S., CARBONELL, J. G. and XING, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics* **6** 719–752.
- [16] COMBETTES, P. L. and WAJS, V. R. (2005). Signal Recovery by Proximal Forward-Backward Splitting. *Multiscale Modeling & Simulation* **4** 1168–1200.
- [17] CUINGNET, R., GERARDIN, E., TESSIERAS, J., AUZIAS, G., LEHERICY, S., HABERT, M.-O., CHUPIN, M., BENALI, H., COLLIOT, O. and ALZHEIMER’S DISEASE NEUROIMAGING INITIATIVE (2011). Automatic classification of patients with Alzheimer’s disease from structural MRI: a comparison of ten methods using the ADNI database. *NeuroImage* **56** 766–781.
- [18] DAUBECHIES, I., DEFRISE, M. and DE MOL, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* **57** 1413–1457.
- [19] DOHMATOB, E., GRAMFORT, A., THIRION, B. and VAROQUAUX, G. (2014). Benchmarking solvers for TV-l1 least-squares and logistic regression in brain imaging. *Pattern Recognition in Neuroimaging (PRNI)*.
- [20] DOHMATOB, E., EICKENBERG, M., THIRION, B. and VAROQUAUX, G. (2015). Speeding-up model-selection in GraphNet via early-stopping and univariate feature-screening. In *PRNI*.
- [21] DUBOIS, M., HADJ-SELEM, F., LÖFSTEDT, T., PERROT, M., FISCHER, C., FROUIN, V. and DUCHESNAY, E. (2014). Predictive support recovery with TV-Elastic Net penalty and logistic regression: An application to structural MRI. In *2014 International Workshop on Pattern Recognition in Neuroimaging* 1–4.
- [22] FRIEDMAN, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of $m \times n$ Rankings. *The Annals of Mathematical Statistics* **11** 86–92. [MR2085](#)
- [23] FRISONI, G. B., FOX, N. C., JACK, C. R., SCHELTENS, P. and THOMPSON, P. M. (2010). The clinical use of structural MRI in Alzheimer disease. *Nature reviews. Neurology* **6** 67–77.
- [24] GRAMFORT, A., THIRION, B. and VAROQUAUX, G. (2013a). Identifying Predictive Regions from fMRI with TV-L1 Prior. In *2013 International Workshop on Pattern Recognition in Neuroimaging (PRNI)* 17–20.
- [25] GRAMFORT, A., THIRION, B. and VAROQUAUX, G. (2013b). Identifying predictive regions from fMRI with TV-L1 prior. In *Pattern Recognition in Neuroimaging (PRNI)*.
- [26] HADJ-SELEM, F., LÖFSTEDT, T., FROUIN, V., GUILLEMOT, V. and DUCHESNAY, E. (2016). Supplementary materials: Iterative Smoothing Proximal Gradient for Regression with Structured Sparsity Research Report, NeuroSpin, CEA, Paris-Saclay, Gif-sur-Yvette - France.
- [27] HARDIN, J., RAMON GARCIA, S. and GOLAN, D. (2013). A method for

- generating realistic correlation matrices. *Annals of Applied Statistics* **7** 1733–1762.
- [28] HE, N., JUDITSKY, A. and NEMIROVSKI, A. (2013). Mirror Prox Algorithm for Multi-Term Composite Minimization and Alternating Directions. *CoRR* **abs/1311.1098**.
 - [29] JACK, C. R., BERNSTEIN, M. A., FOX, N. C., THOMPSON, P., ALEXANDER, G., HARVEY, D., BOROWSKI, B., BRITSON, P. J., L WHITWELL, J., WARD, C., DALE, A. M., FELMLEE, J. P., GUNTER, J. L., HILL, D. L. G., KILLIANY, R., SCHUFF, N., FOX-BOSETTI, S., LIN, C., STUDHOLME, C., DECARLI, C. S., KRUEGER, G., WARD, H. A., METZGER, G. J., SCOTT, K. T., MALLOZZI, R., BLEZEK, D., LEVY, J., DEBBINS, J. P., FLEISHER, A. S., ALBERT, M., GREEN, R., BARTZOKIS, G., GLOVER, G., MUGLER, J. and WEINER, M. W. (2008). The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. *Journal of magnetic resonance imaging: JMRI* **27** 685–691.
 - [30] JACOB, L., OBOZINSKI, G. and VERT, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09* 1–8. ACM Press, New York, New York, USA.
 - [31] LOFSTEDT, T., GUILLEMOT, V., FROUIN, V., DUCHESNAY, E. and HADJ-SELEM, F. Simulated Data for Linear Regression with Structured and Sparse Penalties Technical Report.
 - [32] MAIRAL, J. (2010). Sparse coding for machine learning, image processing and computer vision PhD thesis, École normale supérieure de Cachan.
 - [33] MAIRAL, J., JENATTON, R., BACH, F. R. and OBOZINSKI, G. R. (2010). Network Flow Algorithms for Structured Sparsity. In *Advances in Neural Information Processing Systems 23* (J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, eds.) 1558–1566. Curran Associates, Inc.
 - [34] MICHEL, V., GRAMFORT, A., VAROQUAUX, G., EGER, E. and THIRION, B. (2011). Total Variation Regularization for fMRI-Based Prediction of Behavior. *IEEE Transactions on Medical Imaging* **30** 1328–1340.
 - [35] NESTEROV, Y. (2004). Smooth minimization of non-smooth functions. *Mathematical Programming* **103** 127–152.
 - [36] NESTEROV, Y. (2005). Excessive Gap Technique in Nonsmooth Convex Minimization. *SIAM Journal on Optimization* **16** 235–249.
 - [37] ORABONA, F., ARGYRIOU, A. and SREBRO, N. (2012). PRISMA: PROximal Iterative SMOothing Algorithm.
 - [38] PALOMAR, D. P. and ELDAR, Y. C. (2009). *Convex Optimization in Signal Processing and Communications*. Cambridge University Press.
 - [39] QIN, Z., SCHEINBERG, K. and GOLDFARB, D. (2013). Efficient block-coordinate descent algorithms for the Group Lasso. *Mathematical Programming Computation* **5** 143–169.
 - [40] RAGUET, H., FADILI, J. and PEYRÉ, G. (2013). A Generalized Forward-Backward Splitting. *SIAM Journal on Imaging Sciences* **6** 1199–1226.
 - [41] SAVCHYNSKY, B., SCHMIDT, S., KAPPES, J. and SCHNORR, C. (2011).

- A study of Nesterov's scheme for Lagrangian decomposition and MAP labeling. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* 1817–1823.
- [42] SAVCHYNSKYY, B., SCHMIDT, S., KAPPES, J. and SCHNOERR, C. (2012). Efficient MRF Energy Minimization via Adaptive Diminishing Smoothing. In *Proceedings of the Twenty-Eighth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-12)* 746–755. AUAI Press, Corvallis, Oregon.
 - [43] SCHMIDT, M., LE ROUX, N. and BACH, F. (2011). Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization. In *NIPS'11 - 25 th Annual Conference on Neural Information Processing Systems*.
 - [44] SILVER, M. and MONTANA, G. (2012). Fast identification of biological pathways associated with a quantitative trait using group lasso with overlaps. *Statistical applications in genetics and molecular biology* **11** Article 7.
 - [45] TRAN-DINH, Q., FERCOQ, O. and CEVHER, V. (2015). A Smooth Primal-Dual Optimization Framework for Nonsmooth Composite Convex Minimization. *arXiv preprint arXiv:1507.06243*.
 - [46] VAROQUAUX, G., EICKENBERG, M., DOHMATOB, E. and THIRION, B. (2015). FFASTA: A fast solver for total-variation regularization of ill-conditioned problems with application to brain imaging. In *Colloque GRETSI*. P. Gonçalves, P. Abry.
 - [47] VOUNOU, M., NICHOLS, T. E. and MONTANA, G. (2010). Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. *NeuroImage* **53** 1147–59.
 - [48] WAHLBERG, B., BOYD, S., ANNERGREN, M. and WANG, Y. (2012). An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems. *ArXiv e-prints*.
 - [49] WHELAN, R., CONROD, P. J., POLINE, J.-B., LOURDUSAMY, A., BANASCHEWSKI, T., BARKER, G. J., BELLGROVE, M. A., BÜCHEL, C., BYRNE, M., CUMMINS, T. D. R., FAUTH-BÜHLER, M., FLOR, H., GALLINAT, J., HEINZ, A., ITTERMANN, B., MANN, K., MARTINOT, J.-L., LALOR, E. C., LATHROP, M., LOTH, E., NEES, F., PAUS, T., RIETSCHER, M., SMOLKA, M. N., SPANAGEL, R., STEPHENS, D. N., STRUVE, M., THYREAU, B., VOLLSTAEDT-KLEIN, S., ROBBINS, T. W., SCHUMANN, G. and GARAVAN, H. (2012). Adolescent impulsivity phenotypes characterized by distinct brain networks. *Nature Neuroscience*.
 - [50] YUAN, M. and LIN, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68** 49–67.
 - [51] ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B* **67** 301–320.